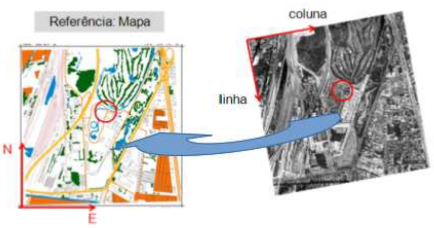
Fundamentos de imagens digitais
02/04/2018

Transformações geométricas no plano

-Exercícios

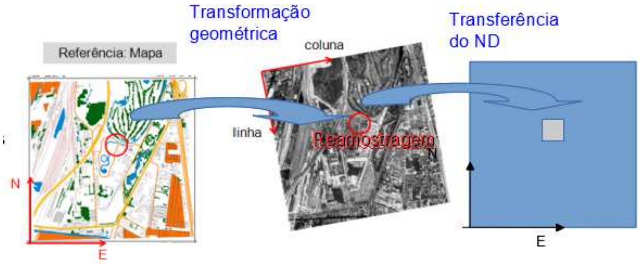
Mapeamento direto:
E = f(linha,coluna)
N = f (linha,coluna)

Cada pixel da imagem é projetado na correspondente posicão na sistema de coordenadas da imagem referencia.

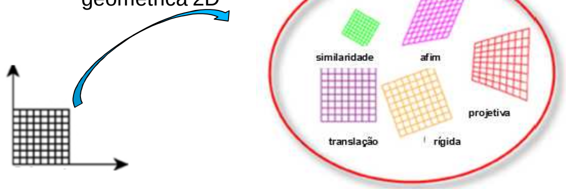


Mapeamento inverso:
linha = f(E,N)
coluna = f (E,N)

Para cada posição (E, N) da referência são calculadas as coordenadas (linha,coluna) na imagem, e é definido o valor (ND) a ser transferido para a posição (E,N) da imagem de saída.



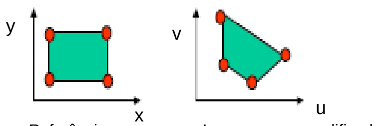
Transformação geométrica 2D



Transformação linear	Fatores modelados	Propriedade característica
Ortogonal ou Rígida (O)	Translação (2) e rotação	Comprimento invariante
Similaridade (S)	Translação (2), rotação e escala	Forma invariante
Afim (A)	Translação (2), rotação, escala (2), e não ortogonalidade	Paralelismo invariante
Projetiva (P)	Centros perspectivos (2)	Colinearidade invariante

Tais fatores são
a)Conhecidos **a priori**, ou
b)São **determinados** (calculados) a partir de pontos comuns (medidos) nos dois sistemas de coordenadas.

Mapeamento inverso



Referência

Imagem a ser modificada

Matricialmente, para n pontos:

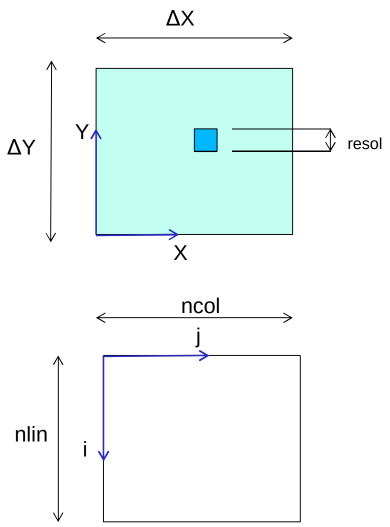
$$\overbrace{\begin{vmatrix} u_1 & u_2 & \dots & u_n \\ v_1 & v_2 & \dots & v_n \\ 1 & 1 & \dots & 1 \end{vmatrix}}^{B} = \overbrace{\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{vmatrix}}^{X} \overbrace{\begin{vmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ 1 & 1 & \dots & 1 \end{vmatrix}}^{A}$$

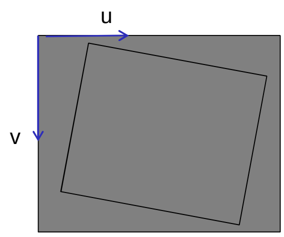
c, f : fatores de translação ;
a,b,d,e : fatores de rotação,
cizalhamento, escala (2 fatores) ;
g,h: centros perspectivos

g = 0 ; h = 0 → Transformações ortogonal, de similaridade e afim

g ≠ 0 ; h ≠ 0 → Transformação projetiva

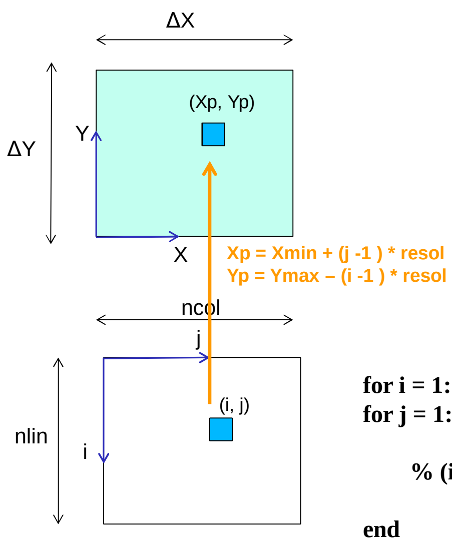
A matriz X resulta de: $\mathbf{X = B \cdot A^T \cdot inv(A \cdot A^T)}$

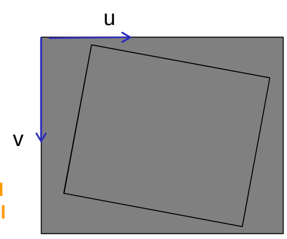




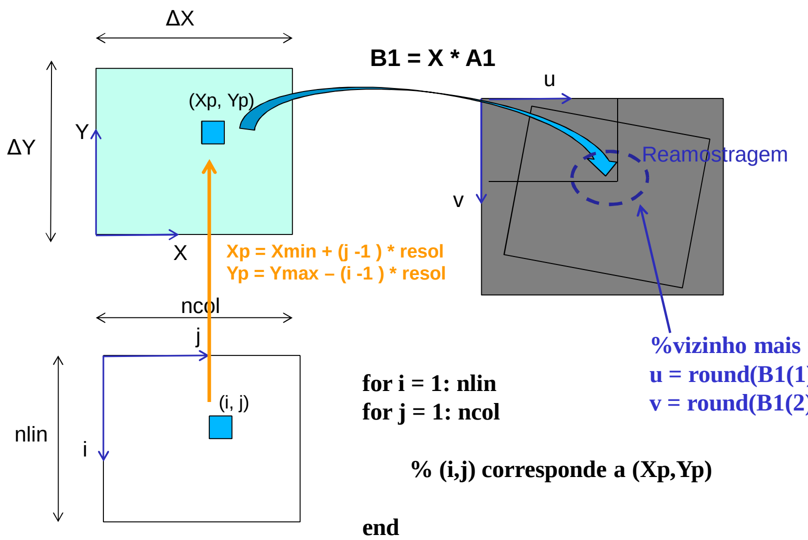
nlin = ceil((Ymax – Ymin)/ resol)

ncol = ceil((Xmax – Xmin)/ resol)



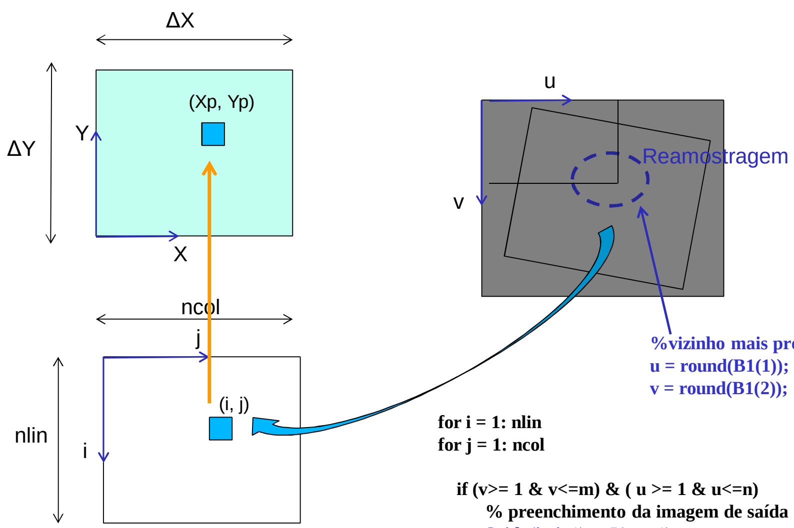



```
for i = 1: nlin
for j = 1: ncol

      % (i,j) corresponde a (Xp,Yp)

end
end
```



%vizinho mais próximo
u = round(B1(1));
v = round(B1(2));

```
for i = 1: nlin
for j = 1: ncol

      % (i,j) corresponde a (Xp,Yp)

end
end
```

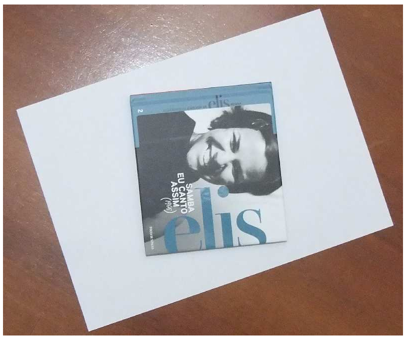


%vizinho mais próximo
u = round(B1(1));
v = round(B1(2));

```
for i = 1: nlin
for j = 1: ncol

   if (v>= 1 & v<=m) & ( u >= 1 & u<=n)
        % preenchimento da imagem de saída
        Saida(i,  j, 1)  = I(v,u, 1);
      Saida(i,  j,2) = I(v,u,2);
        Saida(i,  j, 3) = I(v,u,3);
   end
end
end
```

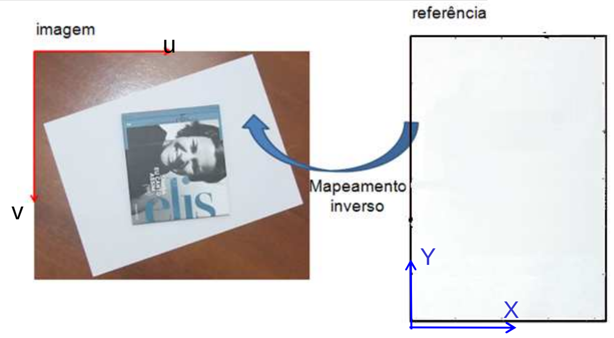
Exercício 1

- **Descrição do Problema:** Deseja-se gerar uma imagem tendo como fundo a folha A4, e com resolução espacial de 0,5 mm.



Número do ponto	u	v	x	y
1	24	209	0	0
2	169	651	210	0
3	777	439	210	297
4	630	8	0	297

Mapeamento inverso



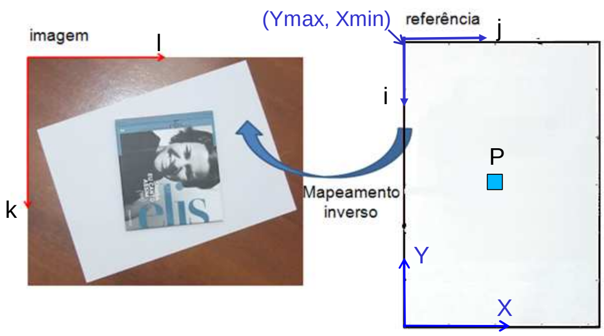
% Coordenadas dos cantos da folha
% no sist. da imagem

```
B = [
 24  169  777  630
209  651  439    8
  1    1    1    1]
```

%Coordenadas dos cantos da folha

```
A = [
0  210  210    0
0    0  297  297
1    1    1    1
]

X = B * A' * inv(A*A')
```

Mapeamento inverso



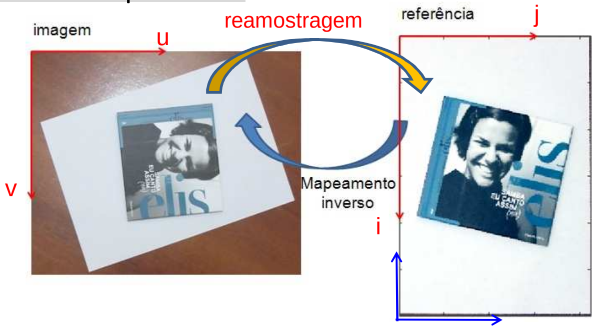
Resolução espacial de 0,5 mm: resol = 0.5

-> quantas linhas e colunas terá a imagem de saída?

nlin =
ncol=

Mapeamento inverso, reamostragem com o método do vizinho mais próximo

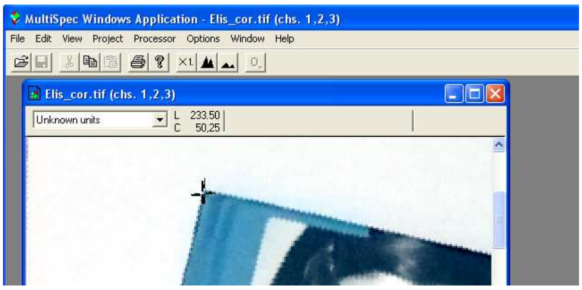
```
I = imread('Elis.tif');
[m,n,k] = size(I);

for i = 1:nlin
for j = 1:ncol

Xp = Xmin + (j -1 ) * resol;
Yp = Ymax - (i -1 ) * resol;
A1 = [ Xp ;  Yp ; 1 ];
B1 = X*A1;

u = round(B1(1));
v = round(B1(2));

if (v>= 1 & v<=m) & ( u >= 1 & u<=n)
        Saida(i, j, 1) = I(v,u,1);
        Saida(i, j, 2) = I(v,u,2);
        Saida(i, j, 3) = I(v,u,3);
end

end
end
imagesc(Saida)
```



- Criando o arquivo o arquivo .tfw:

0.5 ← Resolução espacial segundo X
0
0 ← Resolução espacial segundo Y
-0.5
0
297.0

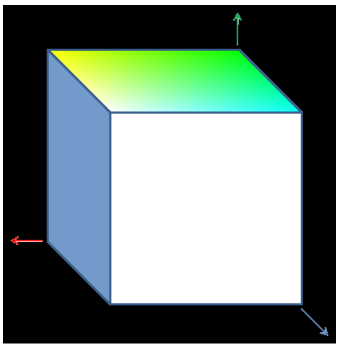
Coordenadas X e Y do canto superior esquerdo da imagem



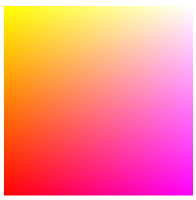
Exercício 2

Construir o cubo de cores RBG, preenchendo os lados faltantes.

cubo.tif



planoB255.tif



planoR255.tif